

湖南大学实训报告

课程名称：____ 计算与人工智能概论 ____ 实验类型：____ 团队实训 ____

实验项目名称：____ 词频统计词云生成 ____

学生姓名：____ 刘家豪 ____ 班级：____ 数学类 2403 ____ 学号：____ 202410040311 ____

同组姓名：____ 陈鸿达 ____ 班级：____ 数学类 2403 ____ 学号：____ 202410040308 ____

一、实训目的

1. 巩固计算与人工智能概念课程所学基础知识
2. 拓展训练学生计算和 AI 思维能力
3. 加强团队分工与合作，培养学生团队协作能力

二、实训内容和团队分工

//1 任务描述：词云又叫文字云，是对文本数据中出现频率较高的“关键词”在视觉上的突出呈现，形成关键词的渲染形成类似云一样的彩色图片，从而一眼就可以领略文本数据的主要表达意思。

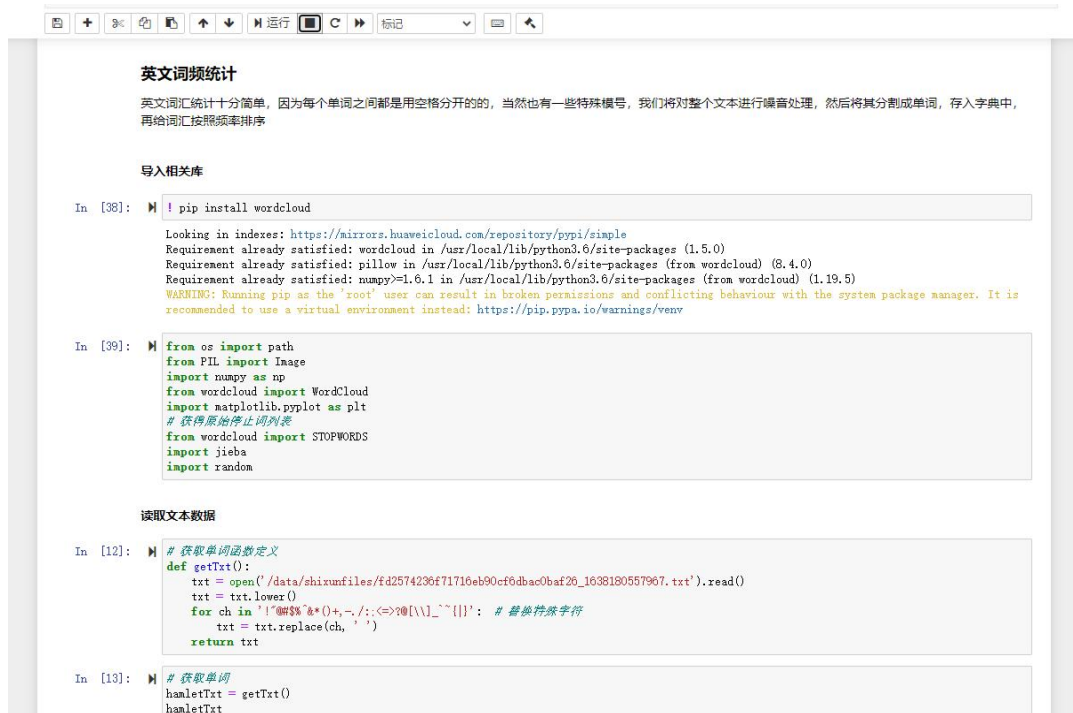
//2 分工：刘家豪负责代码的初步获取及删减；陈鸿达负责对代码进行进一步修改及测试。

三、实训环境

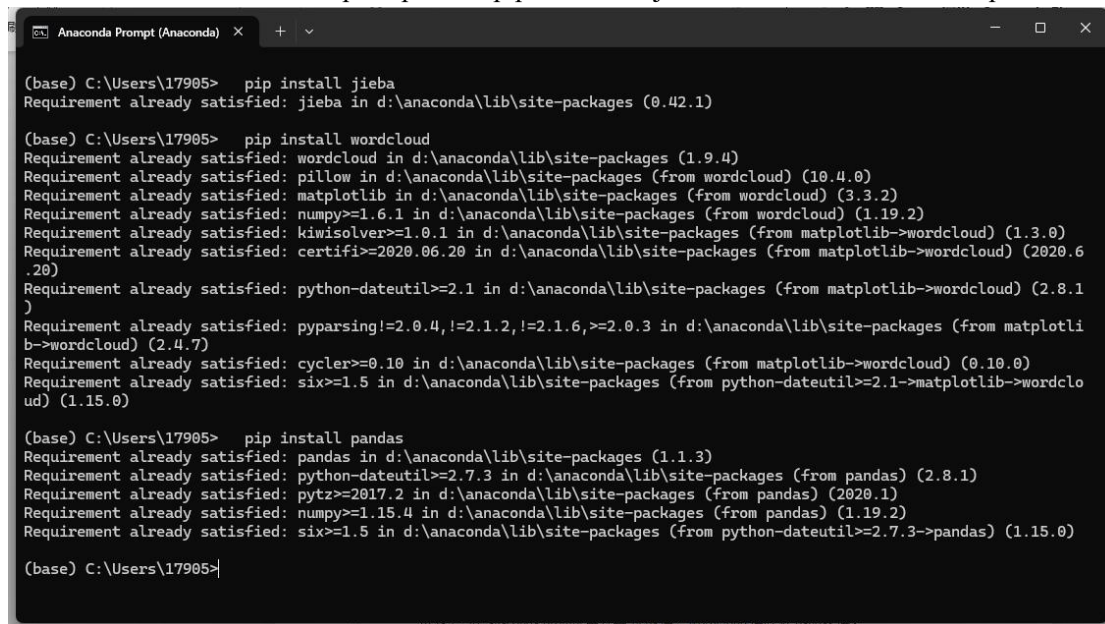
使用 **anaconda** 中的 **jupyter notebook** 创建 **python 3** 文件，使用 **jieba** 库进行分词，**wordcloud** 库生成词可视化。

四、实训方法和步骤

步骤 1、在头歌的实训任务中获取代码，并根据实际需要进行删改。



步骤 2、使用 anaconda prompt 中的 pip 指令安装 jieba 库、wordcloud 库以及 pandas 库。



步骤 3、使用事先准备好的文档以及图片进行测试，并增加可让用户输入文件目录的代码。

```
input_language = input("1:中文,2:英文")
input_name = input("输入文本文件路径").strip(' ') # 去除双引号
input_encoding = input("输入文本编码方式")
input_mask = input("输入图片掩膜路径").strip(' ') # 去除双引号

# 规范化路径
input_name = os.path.normpath(input_name)
input_mask = os.path.normpath(input_mask)
```

五、实训结果和分析

代码主要部分如下：

```
input_language = input("1:中文,2:英文")
input_name = input("输入文本文件路径").strip(' ') # 去除双引号
input_encoding = input("输入文本编码方式")
input_mask = input("输入图片掩膜路径").strip(' ') # 去除双引号

# 规范化路径
input_name = os.path.normpath(input_name)
input_mask = os.path.normpath(input_mask)

if input_language == "2":
    def getTxt():
        txt = open(input_name, encoding=input_encoding).read()
        txt = txt.lower()
        for ch in '!@#%&*()+,.-/:;<=>?@[\\]_`~{|}':
            txt = txt.replace(ch, ' ')
        return txt

    textTxt = getTxt()
    words = textTxt.split()
    # 切割为列表格式
    words = textTxt.split()
    words

    # 遍历统计
    counts = {}
    for word in words:
        counts[word] = counts.get(word, 0) + 1
    counts

    # 原始停止词
    STOPWORDS

    # 建立排除库, 排除掉大多数冠词、代词、连接词等语法型词汇
    for word in list(STOPWORDS):
        # 根据停留词进行排除, 没有找到则返回0
        counts.pop(word, 0)

    # 转换类型
    items = list(counts.items())

    # 按次数从大到小排序
    items.sort(key = lambda x:x[1], reverse = True)
    items
    infos, counts = [], []
    for i in range(10):
        word, count = items[i]
        infos.append(word)
        counts.append(count)
        print(' {0:<10} {1:>5}'.format(word, count))

    # 绘制柱状图进行可视化
    plt.bar(range(len(infos)), counts, width=0.8)
    plt.xticks(list(range(0, 10)), infos, fontsize=12)
    for a, b in zip(np.arange(len(infos)), counts):
        plt.text(a, b, '%d' % b, ha='center', va='bottom', fontsize=12)
    plt.show()

    # 读取文件
    text_y = open(input_name, encoding=input_encoding).read()
    # 读取图片数据
    photo_mask = np.array(Image.open(input_mask))
    # 定义词云图, 背景为白色, 设置掩膜图片
    # max_words:要显示的词的最大个数
    wordcloud = WordCloud(background_color="white", max_words=2000, mask=photo_m
    # 根据文件生成词云
    ax = wordcloud.generate(text_y)
    width,height = 24, 14
    # 默认画布大小
    plt.figure()
    plt.figure(figsize=(width,height))

    # 显示词云
    plt.imshow(ax, interpolation='bilinear')
    plt.axis('off')
    plt.show()
    # 保存图片
    wordcloud.to_file("yingwenciyun.png")
```

```

if input_language=="1":
    # 设置全局中文字体为微软雅黑
    plt.rcParams['font.sans-serif'] = ['Microsoft YaHei']
    plt.rcParams['axes.unicode_minus'] = False # 解决负号显示问题

    # 读取文本
    with open(input_name, 'r', encoding=input_encoding) as f:
        text_z = f.read()

    # 分词并生成列表 (用于统计词频)
    text_list = list(jieba.cut(text_z))

    # 统计词频
    count_ = {}
    for word in text_list:
        if word.strip():
            count_[word] = count_.get(word, 0) + 1

    # 加载停用词
    with open(r'chstopwords.txt', 'r', encoding='utf-8') as file:
        zh_tc = set(line.strip() for line in file)

    # 过滤停用词
    count_ = {word: freq for word, freq in count_.items() if word not in zh_tc}

    # 将词频转换为排序后的列表
    items = sorted(count_.items(), key=lambda x: x[1], reverse=True) # 关键修复点

    # 提取前10个高频词
    infos, counts = [], []
    for i in range(10):
        word, count = items[i]
        infos.append(word)
        counts.append(count)
        print(f' {word:<10} {count:>5}')

    # 绘制柱状图
    plt.bar(range(len(infos)), counts, width=0.8)
    plt.xticks(list(range(0, 10)), infos, fontsize=12, fontproperties='SimHei')
    for a, b in zip(np.arange(len(infos)), counts):
        plt.text(a, b, f' {b}', ha='center', va='bottom', fontsize=12)
    plt.show()

    # 设置图片掩膜
    mask = np.array(Image.open(input_mask))

    # 颜色函数
    def random_color(word, font_size, position, orientation, font_path, random_state):
        s = 'hsl(0, %d%%, %d%%)' % (random.randint(60, 80), random.randint(60, 80))
        return s

    wc = WordCloud(color_func=random_color, font_path=r"C:\Users\10286\Desktop\big work\SimHei.ttf",

    width,height = 24, 14
    # 默认画布大小
    plt.figure()
    plt.figure(figsize=(width,height))
    # 显示词云
    plt.imshow(wc, interpolation='bilinear')
    plt.axis('off')
    wc.to_file("zhongwenciyun.png")

```

注：由于文档编码格式不同，所以在用户输入文档位置后仍需输入此文档的编码格式，以防程序无法正确读取文件。

六、讨论与心得

- 1、学习了使用 python 进行词频统计和创建词云，对 python 有了进一步的了解。
- 2、在本次实验的代码测试中，发现文档会有多种编码格式，且无法用其他编码格式打开，从而导致错误；而图片无论是 jpg 还是 png 等格式，均可使用 utf-8 打开。
- 3、代码执行所需时长与文档长度有关，一般生成词云需要 1 分钟左右时间。